

Amendments to the Claims:

This Listing of Claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

Claim 1 (previously presented): A method of converting a source code containing a plurality of instructions in a predetermined order, including new instructions, to object code for use by a processor, the method including the steps of:

copying plurality of instructions to a temporary file, the new instructions each being copied as data in the form of object code corresponding to such instruction; and

applying the plurality of instructions of the temporary file to an assembler to produce object code corresponding to the old instructions and the data forming object code for the new instructions.

Claim 2 (currently amended): A method of assembling source code containing existing machine language instructions and new machine language instructions with an existing assembler to produce object code having machine language instructions corresponding to each of the instructions of the existing instruction set and the new instructions, the method including the steps of:

copying each of the existing machine language instructions to a temporary file;

copying each new machine language instruction to the temporary file as a data directive having a form corresponding to object code corresponding to such new machine language instruction; and

assembling the machine language instructions and the data directives to produce the object code.

Claim 3 (previously presented): A method of compiling source code having a plurality of first instructions and a at least one second instruction with a compiler capable of deciphering the first instructions but not the second instruction, including the steps of:

copying each of the first instructions to a temporary file;
converting the second instruction to an object code equivalent that forms an argument of a predetermined compiler statement that is written to the temporary file in place of the second instruction;
applying the temporary file to the compiler to
convert each of the first instructions to object code equivalents that are written to an object file; and
removing the argument of the predetermined statement to write the argument to the object file.

Claim 4 (currently amended): The method of claim 3, wherein the first instructions and the second instructions instruction are in a predetermined order in the source code, and the predetermined order is maintained when the first instructions and the ~~include~~ predetermined statement corresponding to the second statement are in the temporary file.

Claim 5 (previously presented): The method of claim 3, wherein the predetermined statement is a data directive.

Claim 6 (previously presented): A processing system operable to compile a source code having a plurality of first instructions and at least one second instruction to produce machine-readable code by copying each of the first instructions to a temporary file; and then,
converting the second instruction to an object code equivalent that forms an argument of a predetermined compiler statement that is written to the temporary file in place of the second instruction;

applying the temporary file to the compiler to convert each of the first instructions to object code equivalents that are written to an object file; and
removing the argument of the predetermined statement to write the argument to the object file.

Claim 7 (previously presented): The processing system of claim 6, wherein the predetermined compiler statement is a data directive statement.

Claim 8 (previously presented): A system for source code compilation to produce machine-readable object code, the source code including a plurality of first instructions and at least one second instruction, the system including:

a preprocessor operating to copy each of the first instructions to a temporary file, the second instruction first being converted to an object code equivalent and placed in an argument of a compiler statement, the compiler statement being written to the temporary file in place of the second instruction;

a compiler that receives the temporary file to produce an object file containing, for each of the first instructions a machine-readable object code equivalent, and for the compiler statement, the object code equivalent.